

## FIȘA DISCIPLINEI

### 1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Babeș-Bolyai Cluj-Napoca
1.2 Facultatea	Facultatea de Matematica și Informatica
1.3 Departamentul	Departamentul de Informatica
1.4 Domeniul de studii	Informatica
1.5 Ciclul de studii	Licenta
1.6 Programul de studiu / Calificarea	Informatica - engleza

### 2. Date despre disciplină

2.1 Denumirea disciplinei	Paradigme și tehnici ale programării paralele						
2.2 Titularul activităților de curs	Conf. Dr. Niculescu Virginia						
2.3 Titularul activităților de seminar	Conf. Dr. Niculescu Virginia						
2.4 Anul de studiu	3	2.5 Semestrul	6	2.6. Tipul de evaluare	colocviu	2.7 Regimul disciplinei	Optional

### 3. Timpul total estimat (ore pe semestru al activităților didactice)

3.1 Număr de ore pe săptămână	4	Din care: 3.2 curs	2	3.3 seminar/laborator	11a b+ 1pr
3.4 Total ore din planul de învățământ	48	Din care: 3.5 curs	24	3.6 seminar/laborator	24
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					30
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					28
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					40
Tutoriat					15
Examinări					14
Alte activități: .....					-
3.7 Total ore studiu individual			127		
3.8 Total ore pe semestru			175		
3.9 Numărul de credite			7		

### 4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Fundamentele programării, Structuri de date
4.2 de competențe	Abilitati de implementare programe.

### 5. Condiții (acolo unde este cazul)

5.1 De desfășurare a cursului	Sala cu proiector
5.2 De desfășurare a seminarului/laboratorului	laborator cu statii de lucru

### 6. Competențele specifice acumulate

C 4.1 Definirea conceptelor și principiilor de bază ale informaticii, precum și a teoriilor și modelelor matematice

C 4.2 Interpretarea de modele matematice și informatice (formale)

C 4.3 Identificarea modelelor și metodelor adecvate pentru rezolvarea unor probleme reale

C 4.4 Utilizarea simulării pentru studiul comportamentului modelelor realizate și evaluarea performanțelor

C 4.5 Încorporarea de modele formale în aplicații specifice din diverse domenii

<b>Competențe profesionale</b>	<ul style="list-style-type: none"> <li>• Fiecare student trebuie să dovedească faptul că a dobândit un nivel de cunoștințe și înțelegere a subiectului și că este capabil (a) să exprime aceste cunoștințe; și, de asemenea, că poate folosi aceste cunoștințe în rezolvarea unor probleme prin implementarea soluțiilor folosind programarea paralelă.</li> <li>○ Abilitatea de a aplica tehnici specifice programării paralele în rezolvarea problemelor.</li> <li>○ Abilitatea de a evalua creșterea de performanță obținută prin folosirea paralelizării.</li> <li>○ Abilitatea de a implementa programe paralele.</li> </ul> <p><b>C4</b></p> <p>C 4.1 Definirea conceptelor și principiilor de bază ale informaticii, precum și a teoriilor și modelelor matematice</p> <p>C 4.2 Interpretarea de modele matematice și informatice (formale)</p> <p>C 4.3 Identificarea modelelor și metodelor adecvate pentru rezolvarea unor probleme reale</p> <p>C 4.4 Utilizarea simulării pentru studiul comportamentului modelelor realizate și evaluarea performanțelor</p>
<b>Competențe transversale</b>	<ul style="list-style-type: none"> <li>- <b>CT1</b> Aplicarea regulilor de muncă organizată și eficientă, a unor atitudini responsabile față de domeniul didactic-științific, pentru valorificarea creativă a propriului potențial, cu respectarea principiilor și a normelor de etică profesională</li> <li>- <b>CT3</b> Utilizarea unor metode și tehnici eficiente de învățare, informare, cercetare și dezvoltare a capacităților de valorificare a cunoștințelor, de adaptare la cerințele unei societăți dinamice și de comunicare în limba română și într-o limbă de circulație internațională</li> </ul>

**7. Obiectivele disciplinei** (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> <li>- Aprofundarea conceptelor de bază ale programării paralele.</li> <li>- Abilitatea de a lucra independent sau în echipă pentru a rezolva probleme într-un context de programare paralelă.</li> <li>- Cunoaștere și abilități de implementare a paradigmei și tehnicilor de bază ale programării paralele.</li> </ul>
7.2 Obiectivele specifice	<ul style="list-style-type: none"> <li>- Cunoașterea și folosirea celor mai importante paradigme/tehnici de dezvoltare a programelor paralele.</li> <li>- Implementarea programei paralele folosind Java, C#, C++, Scala.</li> <li>- Realizarea unui Proiect (aplicație) folosind pentru implementare una dintre variantele specificate anterior</li> </ul>

**8. Conținuturi**

8.1 Curs	Metode de predare	Observații
1. Programarea paralelă: generalități • Paralelism implicit vs. Paralelism explicit - Modelul Data-parallel - Modelul Message-passing - Modelul Shared-variable - Arhitecturi paralele – tendințe actuale	Expunere, descriere, explicație, exemple, discuții ale unor studii de caz.	
2. Pipeline ca paradigma de programare paralelă. Exemple – implementări concrete	Expunere, descriere, explicație, exemple, discuții ale unor studii de caz.	
3. Streaming programming Java streams. Comparatie cu sabloanele pipeline și divide & impera	Expunere, descriere, explicație, exemple, discuții ale unor studii de caz.	

4. Sabloane - <i>Task-Farm/Work-Pool/Thread-Pool</i> Exemple; Java Executors; Fork-Join Pool; Work-stealing; Futures	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
5. Modelul PRAM (Parallel Random Access Machine)	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
6. Retele computationale. Exemplificari. Evaluarea performantei programelor paralele Scalabilitate. Evaluare teoretica si experimentală. - Tehnica reducerii ciclice par-impair	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
7. Descompunerea geometrica a domeniului de date Algoritmi paraleli din algebra liniara. Exemple de algoritmi paraleli din teoria grafurilor. - Tehnica parcursurului Eulerian	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
8. Tehnici utilizate in constructia algoritmilor paraleli - Tehnica dublării recursive - Tehnica par-impair Algoritmi de sortare; (merge sort, quicksort); odd-even sort, counting sort; Bitonic sort	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
9. Hybrid approaches: MPI + threads, MPI+OpenMP	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
10. Modelul Actor	Expunere, descriere, explicatie, exemple.	
11. Prezentare proiecte	Expunere, descriere, explicatie, exemple, discutii ale unor studii de caz.	
12. Prezentare proiecte	Discutii, analize, evaluari.	

#### Bibliografie

1. A Pattern Language for Parallel Programming. Berna L. Massingill, Timothy G. Mattson, and Bever Sanders, Addison Wesley Software Patterns Series, 2004.
2. Michael McCool, Arch Robinson, James Reinders, Structured Parallel Programming: Patterns for Efficient Computation,” Morgan Kaufmann, 2012 .
3. Ian Foster. Designing and Building Parallel Programs, Addison-Wesley 1995.
4. Grama, A. Gupta, G. Karypis, V. Kumar. Introduction to Parallel Computing, Addison Wesley, 2003.
5. D. Grigoras. Calculul Paralel. De la sisteme la programarea aplicatiilor. Computer Libris Agora, 2000.
6. V. Niculescu. Calcul Paralel. Proiectare si dezvoltare formala a programelor paralele. Presa Univ. Clujana, 2006.
7. D.B. Skillicorn, D. Talia. Models and Languages for Parallel Computation. ACM Computer Surveys, 30(2) pg.123-136, June 1998.
8. B. Wilkinson, M. Allen, Parallel Programming Techniques and Applications Using Networked Workstations and Parallel Computers, Prentice Hall, 2002
9. E.F. Van de Velde. Concurrent Scientific Computing. Spring-Verlag, New-York Inc. 1994.

8.2 Seminar / laborator	Metode de predare	Observații
1. Alegerea și specificarea unei probleme corespunzătoare proiectului	Dialog, dezbateri, explicație, exemple.	
2. Proiectarea și implementarea soluției secvențiale	Dialog, dezbateri, explicație, exemple.	
3. Proiectarea soluției paralele. Alegerea unui șablon de programare paralelă adecvat. Evaluare teoretică a soluției paralele	Dialog, dezbateri, explicație, exemple.	
4. Implementarea soluției paralele.	Dialog, dezbateri, explicație, exemple.	
5. Testare și validare	Dialog, dezbateri, explicație, exemple.	
6. Prezentări ale proiectelor studenților	Dialog, dezbateri, explicație, exemple.	
<b>Bibliografie</b> <ol style="list-style-type: none"> <li>1. Parallel Programming in the .NET [<a href="https://msdn.microsoft.com/en-us/library/dd460693(v=vs.110).aspx">https://msdn.microsoft.com/en-us/library/dd460693(v=vs.110).aspx</a>]</li> <li>2. Scala Programming Language, [<a href="http://www.scala-lang.org">http://www.scala-lang.org</a>]</li> <li>3. Java Programming Language [<a href="http://www.oracle.com/us/technologies/java/overview">www.oracle.com/us/technologies/java/overview</a>]</li> <li>4. OpenMP [<a href="http://openmp.org/wp/">http://openmp.org/wp/</a>]</li> <li>5. MPI [<a href="http://www.mpi-forum.org/">http://www.mpi-forum.org/</a>]</li> </ol>		

### 9. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori reprezentativi din domeniul aferent programului

Cursul respectă Recomandarile IEEE and ACM Curricula pentru studii în Computer Science; Cursuri cu tematică similară există în programele de studii ale majorității universităților din țară și străinătate.

Cursul definește dobândirea unor abilități care constituie avantaje evidențiate de potențialele firme angajatoare din domeniu.

### 10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	În scopul evaluării obținerii competențelor specifice, fiecare student va dezvolta un proiect prin care va rezolva o problemă cu specificații date, folosind programare paralelă (se va alege unul dintre limbajele/tehnologiile discutate). Proiectul va avea pe lângă implementare și o documentație care va descrie metoda de paralelizare și evaluarea performanței.	colocviu	Proiect: -specificarea problemei și implementarea secvențială – 15%
10.5 Laborator			- proiectare paralelă și evaluare teoretică a performanței – 20%  - Implementare și testare și analiză performanță - 45%  - prezentare proiect și întrebări legate de deciziile de proiectare și implementare, analiză critică a altor proiecte - 25%

10.6 Standard minim de performanță

Minim 5 la media finala.

Data completării

..15.04.2020...

Titular de curs

.....Niculescu Virginia.....

Titular de laborator

.....Niculescu Virginia.....

Data avizării în departament

.....

Director de departament

.....